PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classificati n 6 :
G06F 17/60, 17/30

A1

(11) International Publication Number: WO 96/34354

(43) International Publication Date: 31 October 1996 (31.10.96)

(21) International Application Number: PCT/US96/05734

(22) International Filing Date: 24 April 1996 (24.04.96)

(30) Priority Data:
08/431,046 28 April 1995 (28.04.95) US

(71) Applicant: UNITED PARCEL SERVICE OF AMERICA, INC. [US/US]; 55 Glenlake Parkway, N.E., Atlanta, GA 30328 (US).

(72) Inventors: COLCHAGOFF, Buff; 935 Pylesville Road, Pylesville, MD 21132 (US). YATES, Carol; 12112 Long Lake Drive, Owings Mills, MD 21117 (US). KONIOR, Louis, R.; 4426 North Woods Trail, Hampstead, MD 21074 (US). SHANK, Clinton, James; 1 Forest Road, Baltimore, MD 21220 (US).

(74) Agents: YOUNG, Jeffrey, E. et al.; Jones & Askew, 37th floor, 191 Peachtree Street, N.E., Atlanta, GA 30303-1769 (US).

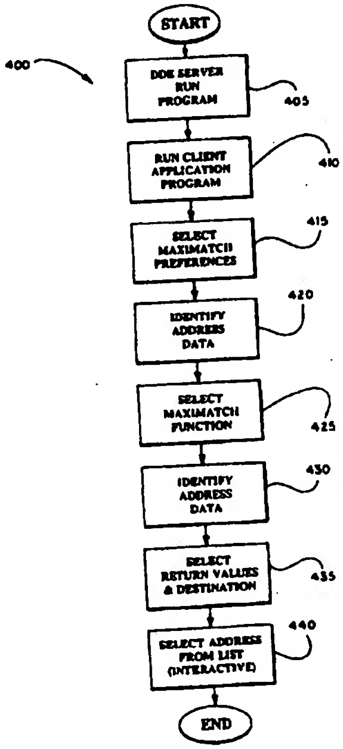
(81) Designated States: CA, JP, European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: SYSTEM AND METHOD FOR VALIDATING AND GEOCODING ADDRESSES

(57) Abstract

A system and method that allow a user to validate and/or geocode address data residing within the user's application program, using a validation and/or geocoding program residing on a remote address validation server. This is accomplished from within the user's application program, such as a spreadsheet, word processor, desktop mapping program, or database management program running on a personal computer, without requiring the user to export and reformat the data. The system utilizes a server application program running on the personal computer, communicating with the user application program via an interprocess communication link such as a DDE utility.


START
400
DDE SERVER RUN PROGRAM
405
RUN CLIENT APPLICATION PROGRAM
410
SELECT MAXIMATCH PREFERENCES
415
IDENTIFY ADDRESS DATA
420
SELECT MAXIMATCH FUNCTION
425
IDENTIFY ADDRESS DATA
430
SELECT RETURN VALUES & DESTINATION
435
SELECT ADDRESS FROM LIST (INTERACTIVE)
440
END

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AM	Armenia
AT	Austria
AU	Australia
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CS	Czechoslovakia
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgystan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakhstan
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam

"SYSTEM AND METHOD FOR VALIDATING AND GEOCODING ADDRESSES"

Technical Field

The present invention relates to a method and system for validating and geocoding addresses, and more particularly relates to such a method and system in which a user may obtain such information from within an application program running on a personal computer.

Background of the Invention

Businesses of various types maintain databases to store the names and addresses of current or potential customers. In some cases, the address data may be unreliable or inaccurate due to clerical errors, errors in data entry, or other problems. There is a need to reduce the chances of misdelivery. In other cases, people ordering goods or services may provide fraudulent addresses in an attempt to defraud a vendor. In order to detect or avoid these problems, businesses utilize reference address databases, such as the United States Postal Services ZIP+4 database to validate addresses.

Many businesses also find it useful to use address data to plot the locations of customers, vendors, or competitors. This may be accomplished by using a geographic coordinate database to obtain the latitude and longitude of the desired address, and to plot those coordinates using a map or mapping computer program. This process is referred to as "geocoding".

Users often store address data in personal computer application programs such as spreadsheet or database management programs. In the prior art, programs that provide geocoding and validation functions have two primary disadvantages. First, in order to validate address data or obtain the corresponding geocodes, the user must export the address data from the application program (e.g., a spreadsheet or database) into a different format that is compatible with the validation or geocoding program. Then the files are sent on a storage medium or uploaded via a bulletin board or directly via a modem to a server computer where validation and geocoding analysis can be performed. The user generally must set up a specific communication protocol to be able to communicate with the server. In the case of sending a disc or uploading to a bulletin board, the user may have to wait one or more days for the data to be returned. Once the requested data is returned by the address validation or geocoding program, it must be reformatted and imported back into the user's application. Thus, obtaining validation or geocoding or address data has been a time consuming and inconvenient process.

The second drawback results from the size of the validation databases. There are over 40 million valid addresses in the United States. Therefore, the validation databases are large and require large amounts of space on computer disk drives. This makes it difficult for businesses to maintain such databases on their own personal computers. Furthermore, the databases must be updated from time to time to ensure that the latest data is available.

Some service providers address the problems of updating and local storage by providing a type of remote batch mode processing as described above. This may require the user to export the files into suitable format and upload the reformatted data to a bulletin board. After the requested data is provided, the data must be downloaded from the bulletin board, reformatted and imported into the user's application program.

Thus, there is a need in the art for a system that allows a user to validate and geocode address data from within the user's application program, without requiring the user to export and reformat the data. Furthermore, there is a need in the art for a system that allows the user to access current address validation and geocoding databases without requiring the data to be exported and reformatted.

Summary of the Invention

The present invention addresses the above-described problems in the art by providing a system and method that allow a user to validate and/or geocode address data residing within the user's application program, using a separate validation and/or geocoding program, which may in some cases reside on a remote address validation server. This is accomplished from within the user's application program, such as a spreadsheet, word processor, desktop mapping program, or database management program, running on a personal computer, without requiring the user to export and reformat the data.

Generally described, the present invention provides a system for validating and/or geocoding addresses stored in a user application program running on a personal computer, comprising: a personal computer configured to run: a multitasking operating system: an interprocess communication mechanism for enabling communication between application programs running under the operating system; a client application program running under the operating system, and capable of storing and operating on stored data representing addresses of persons, entities, and structures; and a server application program running under the operating system, the server application program including: means for generating commands required for controlling the address validation and/or geocoding program; means for establishing an interprocess communication link between the server application program and the client application program; means for detecting a request for address validation and/or geocoding from the client application program via the interprocess communication link; means for receiving address data from the client application program via the interprocess communication link and for preparing the data for processing by the address validation and/or geocoding program; means for establishing communication with an address validation and/or geocoding program operative to match input address data to valid address data stored in a validation database and to output any valid matching address data and/or geocoding values for each input address and for transmitting commands and said data to said address validation and/or geocoding program and for receiving return valid matching address data therefrom via said means for establishing communication with said address validation program and/or geocoding program; and means for sending the valid matching address data and/or geocoding values to the client application program via the interprocess communication link in a form directly usable by the client application program.

The client application program may be, for example, an off-the-shelf spreadsheet program, database management program, word processing program, or desktop mapping program. The interprocess communication link preferably comprises a Dynamic Data Exchange (DDE) utility. The address validation and/or geocoding program may reside on a remote computer, and communication between the personal computer and the remote computer may be direct or by way of modem communication with a gateway computer connected to the remote computer by a local area network. In the alternative, the three programs, the client application program, the server application program, and the address validation and/or geocoding program, may reside on the same computer of any type capable of performing the functions described herein.

The system preferably has the capability to geocode addresses as well as validate the addresses, or may have only geocoding or only address validation capability.

The present invention also provides a method for validating addresses stored in a user application program running on a personal computer, comprising the steps of: running a client application program on a personal computer under a multitasking operating system, and storing under the client application program data representing addresses of persons, entities, and structures; running a server application program on the personal computer under the multitasking operating system; establishing an interprocess communication link between the server application program and the client application program; detecting a request for address validation and/or geocoding from the client application program via the interprocess communication link; receiving address data from the client application program via the interprocess communication link; establishing communication with an address validation and/or geocoding program; transmitting commands and the data to the address validation and/or geocoding program in a form usable thereby; receiving from said address validation and/or geocoding program any valid matching address data and/or geocoding values for each input address; and sending the valid matching address data and/or geocoding values to the client application program via the interprocess communication link in a form directly usable by the client application program.

Thus, it is an object of the present invention to provide an improved system and method for validating and geocoding addresses, that is easy to use from within a user application program, such as a spreadsheet, word processor, desktop mapping program, or database management program.

Other objects, features and advantages of the present invention will become apparent upon examining the following description of preferred embodiments of the invention, when taken in conjunction with the drawings and the appended claims.

Brief Description of the Drawings

Fig. 1 is an illustration of a personal computer suitable for use in implementing the present invention, and a remote address validation server.

Fig. 2 is a block diagram showing the principal components of the processor chassis in the personal computer of Fig. 1.

Fig. 3 is a block diagram illustrating the interface between a computer's input/output devices, an operating system, and application programs.

Fig. 4 is a flow diagram illustrating the method of the present invention as perceived by a user.

Fig. 5 is a state diagram illustrating the method of the present invention as implemented in a dynamic data exchange (DDE) client application program running on a personal computer.

Fig. 6 is a state diagram illustrating the method of the present invention as implemented in a DDE server application program running on a personal computer.

Fig. 7 is a state diagram illustrating the method of the present invention as implemented in a computer program running on a gateway computer communicating with the personal computer and with an address validation server.

Fig. 8 is a state diagram illustrating the method of the present invention as implemented in a computer program running on a remote address validation server.

Fig. 9 is a representation of a window presented to the user when the generic DDE server application program is run.

Fig. 10 is a representation of a window presented to the user when "Preferences" is selected from the pull down menu shown in Fig. 9.

Fig. 11 is a representation of a window presented to the user when the "Communications" button is clicked in the window of Fig. 10.

Fig. 12 is a representation of a window presented to the user when the "Filters" button is clicked in the window of Fig. 10.

Fig. 13 is a representation of a window presented to the user when "Geocode" is selected from the pull down menu shown in Fig. 9.

Fig. 14 is a representation of a window presenting returned candidates to the user in the interactive mode.

Fig. 15 is a representation of a window presenting a scorecard to the user upon completion of address processing.

Detailed Description of the Preferred Embodiment

Turning first to the nomenclature of the specification, the detailed description which follows is represented largely in terms of processes and symbolic representations of operations performed by conventional computer components, including a central processing unit (CPU), memory storage devices for the CPU, and connected pixel-oriented display devices. These operations include the manipulation of data bits by the CPU and the maintenance of these bits within data structures resident in one or more of the memory storage devices. Such data structures impose a physical organization upon the collection of data bits stored within computer memory and represent specific electrical or magnetic elements. These symbolic representations are the means used by those skilled in the art of computer programming and computer construction to most effectively convey teachings and discoveries to others skilled in the art.

For the purposes of this discussion, a process is generally conceived to be a sequence of computer-executed steps leading to a desired result. These steps generally require physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical, magnetic, or optical signals capable of being stored, transferred, combined, compared, or otherwise manipulated. It is conventional for those skilled in the art to refer to these signals as bits, values, elements, symbols, characters, terms, objects, numbers, records, files or the like. It should be kept in mind, however, that these and similar terms should be associated with appropriate physical quantities for computer operations, and that these terms are merely conventional labels applied to physical quantities that exist within and during operation of the computer.

It should also be understood that manipulations within the computer are often referred to in terms such as adding, comparing, moving, etc. which are often associated with manual operations performed by a human operator. It must be understood that no involvement of a human operator is necessary or even desirable in the present invention. The operations described herein are machine operations performed in conjunction with a human operator or user that interacts with the computer. The machines used for performing the operation of the present invention include general purpose digital computers or other similar computing devices.

In addition, it should be understood that the programs, processes, methods, etc. described herein are not related or limited to any particular computer or apparatus. Rather, various types of general purpose machines may be used with programs constructed in accordance with the teachings described herein. Similarly, it may prove advantageous to construct specialized apparatus to perform the method steps described herein by way of dedicated computer systems with hard-wired logic or programs stored in nonvolatile memory, such as read only memory.

Referring now the drawings, in which like numerals represent like elements throughout the several figures, the present invention and the preferred operating environments will be described.

The Operating Environment

Figs. 1–3 illustrate various aspects of the computing environment in which the present invention is designed to operate. Those skilled in the art will immediately appreciate that Figs. 1–3, and the associated discussion, are intended to provide a brief, general description of the preferred computer hardware, operating system, and application programs, and that additional information is readily available in the appropriate programming manuals, user's guides, and similar publications.

Figs. 1 and 2 illustrate a conventional IBM-compatible personal computer **10** suitable for implementing the present invention. As shown in Fig. 1, the personal computer **10** includes a processor chassis **15**, which houses a mother board and other printed circuit boards (not shown) of the type conventionally used in a personal computer. Fig. 1 also shows the personal computer **10** connected to a gateway computer **16** via a communications link such as a telephone line **18**. The gateway **16** is in turn connected to a remote address validation server **17** via a local area network **19**, preferably by Novell named pipes. In the preferred system, the remote address validation server **17** contains extensive databases and application programs that are used for address geocoding or validation. The present invention may be utilized with a variety of such validation and geocoding programs, such as those provided or maintained by United Parcel Service ("LookUPS system"), Qualitative Marketing Software, Inc. ("Star Data"), and Pitney Bowes. These programs typically include a reference database containing known valid addresses for an area of interest. The address databases may store ranges of addresses and their parity (odd or even side of street) rather than individual addresses. For example, an entry may be "201-209 Elm St." with odd parity.

The mother board of the personal computer **10**, which is illustrated in the block diagram of Fig. 2, includes a central processing unit (CPU) **20**, such as the 80486 or "PENTIUM" microprocessors manufactured by Intel. The mother board also includes read only memory (ROM) 25 and random access memory (RAM) **30**, which are connected to the CPU by the data bus **35**. Also connected to the CPU **20** via the data bus **35** are a display interface **40**, keyboard interface **45**, mouse interface **50**, and a hard drive/floppy drive interface **55**. An internal modem **57** allows the personal computer **10** to communicate with the gateway **16** via the telephone line **18**. Although many other internal components of the processor chassis **15** are not shown, those of ordinary skill in the art will appreciate that such components and the interconnection between them are well known. Accordingly, additional details concerning the internal construction of the personal computer **10** need not be disclosed in connection with the present invention.

Referring again to Fig. 1, the personal computer **10** includes a floppy drive **60** and a hard drive **65**, which are connected to the CPU via the hard drive/floppy drive interface **55** (Fig. 2), for use in reading data from and writing data to magnetic media. The personal computer displays data of various types on a display **70**. A user enters commands and information into the personal computer **10** by using a keyboard **75** and/or a pointing device such as a mouse **80**. Other types of pointing devices include a track pad, track ball or other device suitable for positioning a cursor on a computer display.

Those skilled in the art will understand that the operating system, application programs and data are provided to the computer via one of its memory devices, which include the hard drive, floppy drive, RAM and ROM. In the preferred computer **10**, the hard drive **65** is used to store data and programs, including the operating system and the application program that embodies the present invention. When the personal computer **10** is turned on or reset, it loads the operating system from the hard drive **65** into the RAM **30**. Once the operating system is loaded into RAM, the CPU **20** executes the operating system code and causes the visual elements associated with the operating system to be displayed on the display **70**. When the application program is opened by a user, the program code and relevant data are read from the hard drive **65** and stored in the RAM **30**.

The preferred embodiment of the present invention is designed to operate with Microsoft Corporation's "WINDOWS" operating system, which provides a graphical user interface. However, it should be understood that the invention can readily be implemented in other graphic operating systems, such as IBM's "OS/2" operating system, and the operating system used in "MACINTOSH" computers manufactured by Apple Computer, Inc. Those skilled in the art will understand that a graphic operating system enables the user to select and manipulate objects using the mouse **80** or other suitable pointing device, thereby minimizing the amount of alphanumeric input the user is required to input via the keyboard **75**.

Typically, a graphic operating system provides icons or objects that can be selected by using the mouse to move the mouse cursor over the desired icon and clicking the appropriate mouse button. In most cases the file associated with an icon may be executed or opened by double clicking on the icon associated with the file. Icons may be moved by dragging them to the desired location.

Those skilled in the art will understand that the gateway **16** may be a personal computer similar to the computer **10**, configured with a modem for communication with the personal computer **10** as well as suitable hardware and software to enable it to communicate with the address validation server computer over Novell named pipes in a local area network. Other protocols known to those skilled in the art could be used in the local area network.

Fig. 3 is simplified block diagram illustrating the interaction between the input/output drivers **300** of the computer **10**, the operating system **305**, a DDE server application program **310** and a client application program **315**.

When the computer is turned on, the CPU fetches instructions from the ROM. These instructions perform a power on self test and load the basic input output system (BIOS), which controls the most basic level of communications between the CPU and the peripheral devices. The CPU then loads the operating system into RAM and runs the operating system.

The operating system (in conjunction with the BIOS and device drivers) provides the basic interface between the computer's resources, the user, and the application program. The operating system interprets and carries out instructions issued by the user. For example, when the user wants to load an application program, the operating system interprets the instruction (e.g., double clicking on the application program's icon) and causes the hard drive to load the program into RAM. Once the application program is loaded into RAM, it is executed by the CPU. In case of large programs, CPU loads various portions of program into RAM as needed.

The operating system also provides a variety of functions or services that allow an application program to easily deal with various types of input/output (I/O). This allows an application program to issue relatively simple function calls that cause the operating system to perform all of the steps required to accomplish various tasks.

The operating system communicates with the application program by sending input and window-management messages to the application program. A "window procedure" in the application program then responds to messages received from the operating system. The window procedures must examine each message from the operating system. At that point, the window procedure carries out some specific action based on the message, or passes the message back to the operating system for default processing.

The "WINDOWS" operating system supports Dynamic Data Exchange (DDE), which is an interprocess communication mechanism that allows two programs to communicate with each other. Two application programs, which are referred to as a "server" and a "client", carry on a DDE conversation by posting messages to each other. A DDE server is a program that provides data to another program (the DDE client).

A DDE conversation is initiated by the client program. The DDE server responds by providing the requested data. The DDE client identifies the type of data it wants using three characters strings called the "application," the data "topic," and the data "item." The application string identifies the server application program. The topic name identifies the desired topic. All DDE servers support at least one topic. The item name identifies the specific data item that is being requested. Within each topic, a DDE server supports one or more data items.

Those skilled in the art will appreciate that the foregoing description of DDE is intended only as a general overview. Additional information pertaining to DDE is readily available in the appropriate programming manuals, user's guides, and similar publications.

In the preferred system, the DDE server application is written in Visual Basic and communicates with a DDE client application program, which may be used by a user for compiling lists of addresses, etc. The address data from the DDE client application is made available to the DDE server application program via DDE. This allows the user to initiate the address validation or geocoding functions from the client application program without reformatting or exporting the data. The DDE server application program communicates with the remote address validation server via the gateway, and transmits the input address data. The requested validation data or geocodes are returned to the DDE server application program, and are made available to the client application program via DDE.

The current implementation of the DDE server application program supports several DDE client applications, including Microsoft Excel, published by Microsoft Corporation, FoxPro, published by Microsoft Corporation, MapInfo, published by MapInfo Corporation, and a generic client provided with the DDE server application program. The generic client may open other database files or import text and use it to create a database.

Those skilled in the art will appreciate that many application programs, including the DDE client applications listed above, support various macro languages or programming languages that allow users or other application programs, such as Visual Basic, to alter or extend some aspects of the client application program. For example, Microsoft Excel supports a macro programming language, and MapInfo supports a programming language called "MAPBASIC". Some of these programming languages, such as Visual Basic, provide high level commands that carry out the DDE related functions. These commands and their associated parameters are described in the appropriate programming manuals, user's guides, and similar publications.

In the context of the present invention, these extensions or macros relating to application programs are referred to generically as "stubs" that run under control of the original application. A stub is written for each client application program, and includes a DDE server application menu, and other features related to the DDE server application. The stubs are installed on the hard drive of the personal computer **10** as separately executable programs. The menu and other features are discussed more completely below in conjunction with the operation of the preferred program.

In summary, the input address data in the client application program **315** is passed to the server application program via DDE messages. The server application program **310** passes the address data to the operating system **305** with instructions indicating that the data is intended for the modem. The operating system **305** transmits the address data to the gateway **16** via the input/output drivers **300**, which drive the modem. The data is in turn transmitted to the address validation server **17** via the local area network. The data returned by the remote address validation server reaches the client application program **315** by traveling in the opposite direction along the same path.

At this point, it should be appreciated that operating systems such as the "WINDOWS" operating system are quite complex and provide a wide variety of services that allow users and application programs to utilize the resources available in the personal computer. Those skilled in the art will be familiar with operating systems and their various features, which include, but are in no means limited to the messages and functions described above. For more comprehensive information regarding the "WINDOWS" operating system and its interaction with application programs, the reader may refer to any of a variety of publications, including the Guide to Programming, which is part of the Microsoft Windows Software Development Kit (published by Microsoft), the Win32 Programmer's Reference (published by Microsoft Press), and Programming Windows 3.1 (published by Microsoft Press), all of which are incorporated herein by reference.

The Preferred Method For Performing Address Geocoding and Validation

Turning now to Figs. 4–8, the preferred method for performing address geocoding and validation will be described. Fig. 4 is a flow diagram illustrating the steps performed by the user in order to validate or geocode address data. Fig. 5 is a state diagram illustrating the method of the present invention as implemented in a DDE client application program running on a personal computer. Fig. 6 is a state diagram illustrating the method of the present invention as implemented in a DDE server application program running on a personal computer. Fig. 7 is a state diagram illustrating the method of the present invention as implemented in a program running on a gateway personal computer positioned to interconnect the personal computer with a remote address validation server. Fig. 8 is a state diagram illustrating the method of the present invention as implemented in a computer program running on a remote address validation server.

Fig. 4 is a flow diagram that illustrates the method of the present invention in terms of the steps carried out by the user. The discussion presented in conjunction with the flow diagram summarizes the steps that are required in order to use the preferred embodiment of the present invention to geocode or validate address data. The discussion also describes the options and parameters that are selected by the user.

The object selection method 400 of the present invention begins at step 405 when the user launches the DDE server application program. This is accomplished by double clicking on the DDE server application icon. When the DDE server application program is launched, it runs in the minimized mode, and provides an icon at the bottom of the display in order to indicate that the program is running. At this point, the DDE server application program does not have an active window associated with it.

At step 410 the user launches the desired DDE client application stub by double clicking on the appropriate icon in the DDE server application group window. The user then opens or creates the file containing the address that will be validated or geocoded. As mentioned above, the preferred DDE server application supports at least four client application programs: Microsoft Excel, FoxPro, MapInfo, and a generic client program. Icons are provided in the DDE server application group window identifying stubs associated with supported client application programs. These icons are duplicates of the original program icons, and are intended to be used only to run the client application stubs with the DDE server application program.

When the client application program stub is run from the DDE server application group window, a client application window 460 is opened as shown in Fig. 9, and a special menu item 462 appears on the menu bar that appears at the top of the active client application window. Normally the name of the DDE server application would be used as the title of the menu item, and for purposes of this specification it will be assumed that the title is "Maximatch," and the menu will be referred to herein as the "Maximatch menu." Those skilled in the art will appreciate that the Maximatch menu appears in addition to the menu items normally provided by the client application program, thereby adding functionality to the client application program. The original functions included in the client application program continue to be available to the user.

At step **415** of Fig. 4 the user sets the preferences that will be used by the DDE server application program. This is accomplished by selecting the "Preferences" item **464** from the Maximatch menu. This causes a DDE message to be sent to the DDE server application asking it to display its "Preferences" window. In response, the DDE server application program displays a preferences window **470**, as shown in Fig. 10, which allows the user to select the preferences that will be used by the address searching programs that run on the remote address validation server and the gateway computer. These selections are entered directly into the DDE server application. Each of the preferences is discussed below.

(1) Communication parameters

The preferences window **470** includes a communications button **472** that allows the user to open a communications window **474**, as shown in Fig. 11. This window prompts the user to set the communications parameters used by the DDE server application program. The communications window allows the user to specify the communications port that the modem is connected to, the type of modem that is connected, and the telephone number that will by dialed by the modem to call the remote gateway **16**. The communications window also allows the user to set the modem parameters baud rate and port.

(2) Notification mode

The preferences window **470** allows the user to choose from options **476** pertaining to the notification mode. If the user selects the "notify upon completion" option, the computer will display a scorecard upon completion of the address processing, as shown in Fig. 15. If the user selects the "show progress" option, the computer will also display a status window that is continuously updated while the DDE server application program is processing addresses.

(3) Validation mode

The validation portion of the preferences window **470** allows the user to specify at **478** whether the addresses will be processed automatically or interactively. If the user selects interactive, the DDE server application program displays, for each input address for which there is a match, a validation panel (Fig. 14) showing the current record and a list of candidate addresses returned after processing on the remote address validation server. The program waits for the user to select the correct address. If the user selects automatic, the address validation or geocoding process runs automatically and returns a single address to the user for each input address for which there is a match. The returned address is identified based on search criteria that are discussed below. In automatic mode, or batch mode, the input addresses are transmitted and processed one-by-one without user intervention until all have been completed or the user selects "cancel" and terminates processing.

(4) Search criteria

The preferences window **470** provides several search criteria at **480** that are set by the user and applied to both automatic and interactive address validation and geocoding. The choices for the search criteria depend on the particular validation or geocoding program running on the address validation server. For example, for the LookUPS validation server, the first of these search criteria is called a quality value. The user specifies a number ranging from 1 to 100. The quality value provides a level of acceptability for the remote address validation server to consider when matching addresses. A quality value of 100 tells the remote address validation server software to search for exact matches. A quality value of 1 would allow consideration of everything. Lowering the quality value results in a broader, but slower search. The default quality value is 80. The algorithm used by the remote address validation server is discussed below in conjunction with Fig. 8.

The user also specifies a candidate value that determines how many possible matches will be retained for consideration by the user (in interactive mode) or the gateway (in automatic mode). For example, this number may range from 1 to 15. Increasing the number results in broader, but slower searches. If the user selected the interactive mode, the maximum number of possible matches displayed to the user is determined by this parameter. The default candidate value is 5.

The preferences window **470** also allows the user to instruct the geocoding function to use "zip centroids" if an address cannot be matched. When zip centroids are used, the geocode function will return the coordinates of the center of the zip code area if no match can be found.

(5) Automatic match criteria

When the automatic mode is selected, the preferences window **470** provides additional parameters at **482** for selection by the user. The automatic match criteria parameters specify how the gateway application chooses the best match when in the automatic mode. Examples of automatic match criteria are:

Exact
All address components of the candidate address must exactly match the input address.

Close
Most address components of the candidate address match the input address; Very probably the same address; Usually denotes the misspelling of the street name.

Possible
Few address components of the candidate address match the input address: A good match, but there is some room for error.

Poor
Any candidate can match.

(6) Filters

In addition to the automatic matching criteria, the preferences window **470** also allows the user to select additional filters that are used to filter out unwanted matches in the automatic mode. Clicking on a "Filters" button **484** in the preferences window opens a window **486**, shown in Fig. 12, for selection of filters. The address range matching filter selected at **488** filters out potential matches that do not satisfy the user's address range accuracy requirements. This is especially useful for geocoding areas with sparse address range data. The choices for the automatic range matching criteria are:

Exact
Street range of candidate must be an exact match to input address.

Close
Street range of candidate must be within 200 of input address.

Nearby
Street range of candidate must be within 500 of input address.

Any
Any candidate address range can be accepted.

When choosing the automatic mode, the user may also activate filters **490** and/or **492** to specify whether the candidate address must match the correct side of the street and/or the original zip code. If "must match correct side of street" is selected, the gateway filters out candidates whose address ranges do not match the input street number's parity (if the database indicates that there is an "even" and "odd" side of the street). If "must match original zip code" is selected, the gateway filters out potential matches that do not match the input address's zip code.

After all of the relevant parameters are selected, the method proceeds to step **420** of Fig. 4, and the user selects the address components that will be used by the remote address validation server program. In the preferred system, this step is only required when the client application program is Microsoft Excel. This allows the user to select the cells that contain the input address data.

At step **425** the user selects the desired function from the Maximatch pull down menu. The user may select "Geocode" or "Validate," resulting in the display of a window **495** by the client application, as shown in Fig. 13 for the case of the generic client.

At step **430**, the user identifies the source of the input address data by selecting fields at area **496** of the window **495**. In the preferred system, this step is not required when the client application program is Microsoft Excel. However, with some client application programs, the user must identify the table that includes the input addresses and the specific fields that hold the address components.

At step **435** the user selects the desired return values from all of the data that will be returned by the gateway. The is done using area **497** of the window **495** in a well known manner. The user also identifies the fields in the client application program in which the return values will be stored.

When geocoding addresses, the possible return values include:

Latitude	11 digit numeric with 6 decimal place precision
Longitude	11 digit numeric with 6 decimal place precision
Method	5 character string; indicates AUTO, ZIP [centroid], or USER (interactive)

Evaluation 5 character string; indicates POOR, POSS, CLOSE, EXACT

Rank 6 digit numeric w/ 2 decimal place precision; indicates quality score

Matched to 55 character string; indicates the specific address the input address was matched to

Reject codes 10 character string describing why the best candidate was not acceptable. The reject codes include:

NODATA	No data was found
DECLINED	User declined a choice (interactive mode only)
R	Rural route address not supported
P	Parity (wrong side of street)
#	Street numbers out of range
D	Directional (street prefix did not match)
N	Street name did not match
T	Street type did not match
d	Directional (street suffix did not match)
C	City name did not match
S	State name did not match
Z	Zip code did not match

When validating addresses, the longitude and latitude return values are not available. After the desired return values are selected, the user clicks on the OK button in the window **495** and the geocoding or validation process begins. If the user has selected the "show progress" preference, the status window will display the status of processing being carried out by the DDE server application program. This process is described below in conjunction with Figs. 5-8.

At step **440** the matching addresses or geocodes with matching addresses are returned by the gateway **16** program. If the user selected the interactive mode, the DDE server application program will present a validation window **499** that includes the current input address and a list of possible candidates, as shown in Fig. 14. The size of the list is controlled by the number of candidates variable described above in conjunction with the preferences. The user may select one of the candidates, which is stored in the designated destination fields, or may cancel consideration, or modify the displayed input address and click the "retry" button in the window **499**. If the user has selected the "show progress" preference, the status window will then continue to be updated to show the number of addresses completed, the percentage matched, and what portion of the addresses remains to be processed. The data sending and return steps are repeated until all of the addresses selected at steps **420** or **430** are processed. If the system is operating in the automatic mode, the return values will be stored in the destination fields. At that point, the method **400** is complete.

Figs. 5-8 are state diagrams illustrating the method of the present invention as implemented in the client application program, the DDE server application program, the gateway program, and the remote address validation server, respectively. The purpose of the state diagrams is to illustrate the interaction between each of the program modules. State diagrams have been chosen because the processes are primarily event driven (rather than process driven).

Fig. 5 is a state diagram illustrating the method **500** of present invention as implemented in a client application program. In the preferred system, the client application program is Microsoft Excel, FoxPro, MapInfo, or the generic client modified by a DDE client application stub. As mentioned above, a stub is a program module that is used to alter an application program. The stub is typically written in a macro language or other programming language supported by the application program, such as MS Visual Basic or MapBasic. In the preferred system, the stub is written to add the Maximatch menu to the menu bar, and to add functionality associated with the DDE server application program. This added functionality is described in conjunction with the state diagram.

The method **500** begins at state **505**. At this point, the client application program has not been launched by the user. When the user goes to the DDE server application group and double clicks on the client application stub's icon, the method proceeds to state **510** and loads the client application stub program. After the client application stub is loaded, at state **515** the menu bar with the Maximatch menu item is displayed.

After the client application program and DDE server application stub are loaded, the method advances to the idle state **520**. In the idle state, the DDE server application functions have not been initiated and the normal functions provided by the client application program are available to the user.

When the user selects the preferences item from the Maximatch menu, the method proceeds to state 525. The client application sends a DDE command string to the server application program indicating that the server application program should display its preferences window. The method 500 then returns to the idle state 520.

When the user selects the Geocode or Validate functions in the Maximatch menu, the method proceeds to state 530 and the client application displays a window 495 that allows the user to specify the source (table, fields, etc.) of the input address data, the return values, and the destination fields for the return values (except in the case of Microsoft Excel). The user may elect to cancel the function from the window 495. After the user supplies this information and clicks "OK" in the window, the method proceeds to state 535, where the client application program uses a DDE link to transmit a "connect" command to the DDE server application, which in response establishes modem communication with the gateway 16. After the "connect" command is sent, the client application waits for "ready" to appear in a status field provided by the DDE server application to indicate that the connection has been established. At state 536 the number of addresses to be processed is also sent to the DDE server application in a separate message.

At state 540 the client application program sends the first address to the DDE server application program. After an input address is sent, the client application program waits for "ready" to appear in a status field provided by the server application program to indicate that the return data is available. This is accomplished using DDE messages to poll the status item on a form posted by the DDE server application program. For example, the program, topic, item for the DDE message are the NAME of the DDE server application, VALIDATE (OR GEOCODE), and STATUS, respectively. The client application retrieves the value of the status item from the DDE server application program in response to the polling message.

When the status item indicates that the return data is available, the method proceeds to state 545 and requests the desired return data from the DDE server application program. This is accomplished using a separate DDE message to request each of the specific items (e.g., latitude, longitude, method, rank, etc.) that were identified by the user in the window 495 displayed at state 530. The DDE server application program makes each piece of requested data available to the client application.

From state **545** the method goes to state **548**. In either the automatic mode or the interactive mode, the client application receives and stores the return data in the designated field. If the client application program has additional addresses to send to the DDE server application program, it returns to state **540** and sends the next address. The method **500** then returns to the idle state **520**.

Fig. 6 is a state diagram illustrating the method **600** of present invention as implemented in a DDE server application program. In the preferred system, the DDE server application program interacts directly with the client application program.

The method **600** begins at state **605**. At this point, the server application program has not been launched by the user. When the user goes to the DDE server application group and double clicks on the DDE server application icon, the method proceeds to state **610** and loads the DDE server application program.

After the DDE server application program is loaded, the method advances to the idle state **615**. In the idle state, the DDE server application functions have not been initiated and the normal functions provided by the client application program are available to the user when it is loaded.

When the user selects the Preferences item from the Maximatch menu, the client application program sends a preferences command to the DDE server application program via DDE. This causes the method to proceed to state **620**, where the DDE server application program displays the preferences window. As described above in conjunction with Figs. 4 and 9-12, the preferences window allows the user to set the parameters that will be used to control the validation or geocoding session. After the preferences are entered, the method returns to the idle state **615**.

The method goes to state **621** after it receives a "connect" message via DDE from the client application. The DDE server application then establishes telephone communication with the gateway **16**. At state **625** parameters including the selected preferences and the nature of the processing request (Geocode or Validate) are received from the client application program. This DDE message is sent by the client application program after the user selects the Geocode or Validate options from the Maximatch menu and selects the return data and destination.

At state **635** the DDE server application program transmits the input address data to the gateway **16**. The DDE server application program sends the current preference settings to the gateway, which stores them. After the first address is transmitted, at state **636** a status window is displayed. At state **640** the DDE server application program receives and stores the matching address data from the gateway. In the interactive mode, the method moves to state **641** and displays the match candidates for the user in a validation window **499**, shown in Fig. 14. An additional text box holds all of the characters of the input address concatenated into a single string. The user may select one of the candidates, whose information is stored in various text boxes of a form ready to be accessed by the client application program. Or, the user may modify the original input address, select "retry" and return to state **635** where the modified input address will be transmitted to the gateway. In the automatic mode, the single matching candidate will be stored in the form.

At this point, it should be appreciated that all of the pieces of available information (text pieces of the candidate addresses and associated return values) pertaining to the matching addresses are provided to the DDE server application program, and are stored in an appropriate item field or text box in a window or form. At this point, the data is available to the client application program. The method goes to state **645** and sets the status item in a field of the form to "ready" to indicate that the data is available. At state **650**, the status window is updated if the user selected the "show progress" option in the preference window.

The method **600** continues to make return data items available to the client application program by returning to state **630** as long as it continues to receive data requests from the client application program. If the client application program indicates that all of the input addresses have been processed, the method disconnects the communications link with the gateway.

If all of the input addresses have been processed, the method proceeds to state **655** and displays a scorecard window **656**, as shown in Fig. 15, showing the number of addresses processed, the number of addresses matched to streets, the number of addresses matched to zip codes, the number of addresses that were unmatched, and the elapsed time. A pie chart or other graphical representation of the number of addresses matched and unmatched may be displayed. The method returns to the idle state **615**.

Fig. 7 is a state diagram illustrating the method **700** of present invention as implemented in a computer program running at the gateway **16**. In the disclosed embodiment, the gateway is connected by modem to the personal computer **10** for communication between the DDE server application program and a gateway application program, the function of which will become apparent from the following description. The gateway application program may be written in an appropriate language such as visual basic.

The method **700** begins at state **702**. At this point, the gateway application program has not been launched. When this is done in a conventional manner, the method proceeds to state **703** and launches the gateway application program. At this point, no communication has been established with the personal computer **10**. At state **705**, the gateway application initializes the modem, and the method proceeds to state **710** and establishes separate connections via the local area network **19** to the validation program and the geocoding application programs running on the address validation server **17**. Then the method proceeds to an idle state **715** and waits for a signal from the DDE server application program.

When a call comes in via the modem from the DDE server application program, the gateway program answers the call at state **720**. Then the method proceeds to state **725** where data consisting of the entered preferences and request type (validate or geocode) is received from the DDE server application program and stored. This is followed at state **730** by address data.

Upon receiving an address, the gateway program detects whether the address is parsed in a manner required by the requested address validation server function (validation or geocoding). If parsing is necessary, this causes the method to proceed to state **735**, at which the gateway program parses the address data. Various parsing programs are readily available. The parsing program identifies the components of an address. Those skilled in the art will understand that the parsing parameters and insertion of delimiters between pieces of text will vary depending upon the needs of the particular validation or geocoding application running on the address validation server.

After parsing, the method proceeds to state **740**, where the parsed pieces of the address and the user's preferences are stored in a data structure suitable for transmission over the Novell named pipes to the address validation server **17**, and acceptable to the application running on the address validation server. At state **745**, the gateway transmits the parsed input address data and preferences to the address validation server **17**. At state **750**, the gateway receives and stores the return matching address data and/or geocoding values, and other return values, from the address validation server. The return values that may be specified by the user are listed above. All candidates that are a match according to the address validation server matching program, operating within the "search criteria" preferences described above, are returned to the gateway.

At state **751**, the candidates are sorted. First, the candidates are assigned the automatic match criterion which they achieved. Next, they are sorted by such criteria, in descending order of exactness of the match. Then a secondary sort is conducted by quality score within candidates having the same automatic match criterion. Then a tertiary sort is conducted by closeness of address range. If the user has selected interactive mode, the method proceeds to state **760**, where the candidate matches and return value are transmitted to the DDE server application program (see state **640** of Fig. 6) for user review.

If the user has selected automatic mode, the method then proceeds to state **755**, at which the return data is processed and filtered by the gateway application. The filters specified in the "Filters" window shown in Fig. 12 are applied to each candidate starting at the top of the sorted list. For example, if the "must match original zip code" filter is selected, the candidate will be discarded if it has a differing zip code. If the "must match correct side of street" filter is selected, the candidate will be discarded if its street number has a parity opposite to the original address. The address range filter is applied according to the level selected to discard the candidate if its address range is not close enough to the range of the original address. The first candidate to pass the tests of the filters will be returned to the DDE server, in the automatic mode.

If no candidates result from an address validation attempt, a reject code from the list described above is returned. This is also the case when geocoding produces no match if "ZIP Centroid" use is not selected. If "ZIP Centroid" use has been selected, such latitude and longitude data is returned with the reject code.

The method then proceeds to state **760** where a record containing the parsed address fields of a candidate match, and fields for other return values described above, is assembled. At state **765**, the gateway transmits this data to the DDE server application running on the personal computer **10** via modem.

Fig. 8 is a state diagram illustrating the method **800** of present invention as implemented in a computer program running at the remote address validation server. As discussed above, the remote address validation server contains the databases that are used for address geocoding and validation. The remote address validation server also performs the geocode and validate functions based on the input data and search parameters provided by the DDE server application program running on the personal computer **10** via the gateway **16**.

In the preferred system, the database used to validate addresses is based on the ZIP+4 database provided by the United States Postal Service. This database includes virtually all of the valid addresses in the United States. The database used to geocode addresses is based on commercially available geocoding data, such as the TIGER database, provided by the United States Government. This database includes geographic coordinate data associated with addresses in the United States. The addresses in these databases may be stored all or in part as ranges of addresses and their parity (odd or even side of street) rather than individual addresses. The present invention does not require access to any individual address data.

The method **800** begins at the idle state **805**. At this point, the remote address validation server **17** is available to receive calls from the gateway **16** over the local area network **19**. When a call is received, the method goes to state **810** and establishes a communications session with the gateway for each of the application programs running on the address validation server. Those skilled in the art will appreciate that the remote address validation server is capable of conducting multiple communications sessions simultaneously. As described above, the remote address validation server will use the preference settings to determine the parameters of the validation or geocode process.

When the remote address validation server receives an address from the gateway **16**, the method goes to state **820**, where the remote address validation server searches the appropriate database (geocode or validation) to find matching addresses. This process is controlled by the search criteria included in the preferences that were set by the user. The searching and matching algorithm (which is not part of the present invention) may be any one of several as described above. One skilled in the art will be able to modify the form of the address data and preference parameters according to the input needs of the particular validation or geocoding program being used.

After the match candidates have been determined, the method proceeds to state **825** and transmits the return data to gateway **16**. This data includes all of the data discussed above in conjunction with the selection of the return values.

If the remote address validation server receives another address from the gateway, the method returns to state **820** and processes the new address. If a message from the gateway instructs address validation server to terminate processing, the method terminates the communication session and returns to the idle state **805**.

The foregoing methods of the present invention may conveniently be implemented in a program module that is based upon the state diagrams of Figs. 5-8. No particular programming language has been indicated for carrying out the various procedures described above because it is considered that the operations, steps and procedures described above and illustrated in the accompanying drawings are sufficiently disclosed to permit one of ordinary skill in the art to practice the instant invention. Moreover, there are many computers and operating systems which may be used in practicing the instant invention and therefore no detailed computer program could be provided which would be applicable to these many different systems. Each user of a particular computer will be aware of the language and tools which are most useful for that user's needs and purposes.

The present invention may be embodied on a single computer in various ways. One way is to use the DDE utility for communication between the program units including the address validation and/or geocoding application. Another way would be by linking the functional units represented in the above embodiment by the DDE server application and the address validation and/or geocoding application, into one executable program group. This would eliminate the need for communication via modem. Those skilled in the art will understand that languages such as Visual Basic include commands such as "Call to External Function" that can be utilized to link program units found in a library such as a "Dynamic Link Library."

The present invention has been described in relation to particular embodiments which are intended in all respects to be illustrative rather than restrictive. For example, although the present invention has been described in the context of a computer running the WINDOWS operating system, those skilled in the art will understand that the principles of the present invention may be applied to, and embodied in, any type of computing device.

Alternative embodiments will become apparent to those skilled in the art to which the present invention pertains without departing from its spirit and scope. Accordingly, the scope of the present invention is defined by the appended claims rather than the foregoing description.

Claims

What is claimed is:

1. A system for validating addresses stored in a user application program running on a personal computer, comprising:

a personal computer configured to run:

a multitasking operating system;

an interprocess communication mechanism for enabling communication between application programs running under said operating system;

a client application program running unuer said operating system, and capable of storing and operating on stored data representing addresses of persons, entities, and structures; and

a server application program running under said operating system, and including:

means for generating commands required for controlling said address validation program;

means for establishing an interprocess communication link between said server application program and said client application program;

means for detecting a request for address validation from said client application program via said interprocess communication link;

means for receiving address data from said client application program via said interprocess communication link and for preparing said data for processing by said address validation program;

means for establishing communication with an address validation program operative to match input address data to valid address data stored in a validation database and to output any valid matching address data for each input address and for transmitting commands and said data to said address validation program and for receiving return valid matching address data therefrom via said means for establishing communication with said address validation program; and

means for sending said valid matching address data to said client application program via said interprocess communication link in a form directly usable by said client application program.

2. The system of Claim 1, wherein said client application program comprises a spreadsheet program.

3. The system of Claim 1, wherein said client application program comprises a database management program.

4. The system of Claim 1, wherein said interprocess communication link comprises a Dynamic Data Exchange (DDE) utility.

5. The system of Claim 1, wherein said server application program further includes:

means for generating commands required for controlling an address geocoding program;

means for detecting a request for address geocoding from said client application program via said interprocess communication link;

means for preparing said address data for processing by said address geocoding program;

means for transmitting geocoding commands and said data to said address validation server computer and for receiving return geocoding values therefrom; and

means for sending said geocoding values to said client application program via said interprocess communication link in a form directly usable by said client application program.

6. A system for geocoding addresses stored in a user application program running on a personal computer, comprising:

a personal computer capable of communicating with a remote computer running a geocoding program operative to match input address data to valid address data stored in a validation database and to output any valid matching address data for each input address, said personal computer configured to run:

a multitasking operating system;

an interprocess communication mechanism for enabling communication between application programs running under said operating system:

a client application program running under said operating system, and capable of storing and operating on stored data representing addresses of persons, entities, and structures; and

a server application program running under said operating system, and including:

means for generating commands required for controlling said address geocoding program;

means for establishing an interprocess communication link between said server application program and said client application program;

means for detecting a request for address geocoding from said client application program via said interprocess communication link;

means for receiving address data from said client application program via said interprocess communication link and for preparing said data for processing by said address geocoding program;

means for establishing communication with said remote computer and for transmitting commands and said data to said remote computer and for receiving return geocoding values therefrom via said means for establishing communication with said remote computer; and

means for sending said geocoding values to said client application program via said interprocess communication link in a form directly usable by said client application program.

7. A method for validating addresses stored in a user application program running on a personal computer, comprising:

running a client application program on a personal computer under a multitasking operating system, and storing under said client application program data representing addresses of persons, entities, and structures;

running a server application program on said personal computer under said multitasking operating system;

establishing an interprocess communication link between said server application program and said client application program;

detecting a request for address validation from said client application program via said interprocess communication link;

receiving address data from said client application program via said interprocess communication link;

establishing communication with an address validation program;

transmitting commands and said data to said address validation program in a form usable thereby;

receiving from said address validation program any valid matching address data for each input address; and

sending said valid matching address data to said client application program via said interprocess communication link in a form directly usable by said client application program.

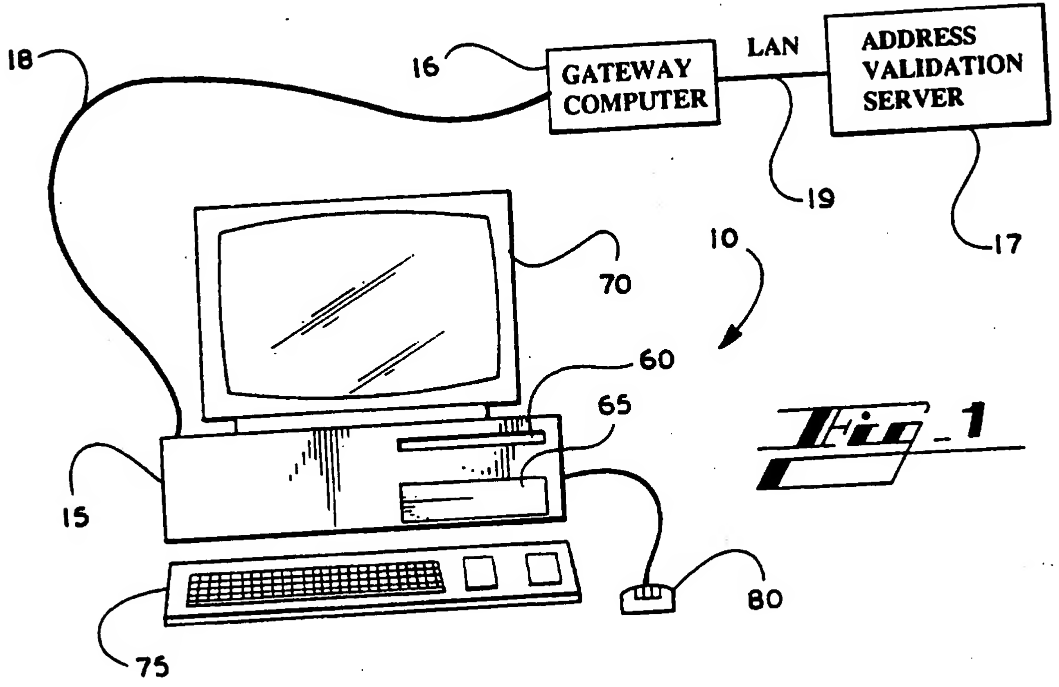
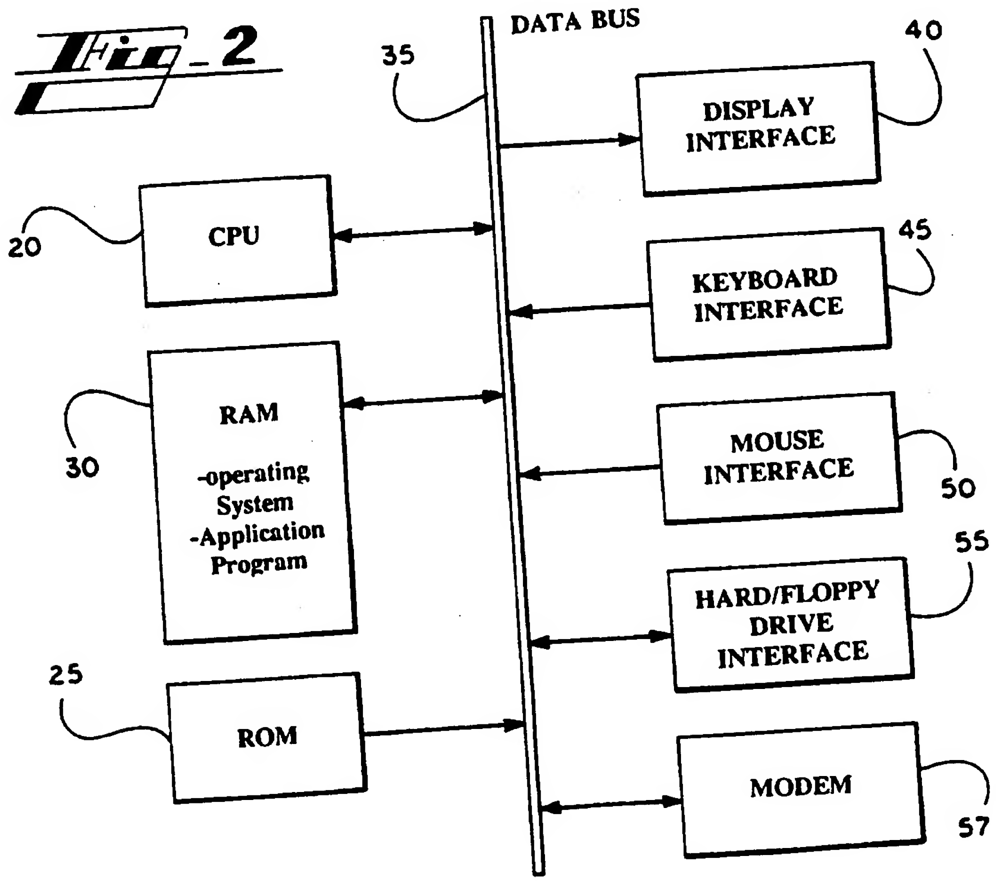

18
16
GATEWAY COMPUTER
LAN
ADDRESS VALIDATION SERVER
19
17
10
70
60
65
15
80
75

Fig_1

Fig_2


DATA BUS
35
40
DISPLAY INTERFACE
20
CPU
45
KEYBOARD INTERFACE
30
RAM
-operating System
-Application Program
MOUSE INTERFACE
50
55
HARD/FLOPPY DRIVE INTERFACE
25
ROM
MODEM
57

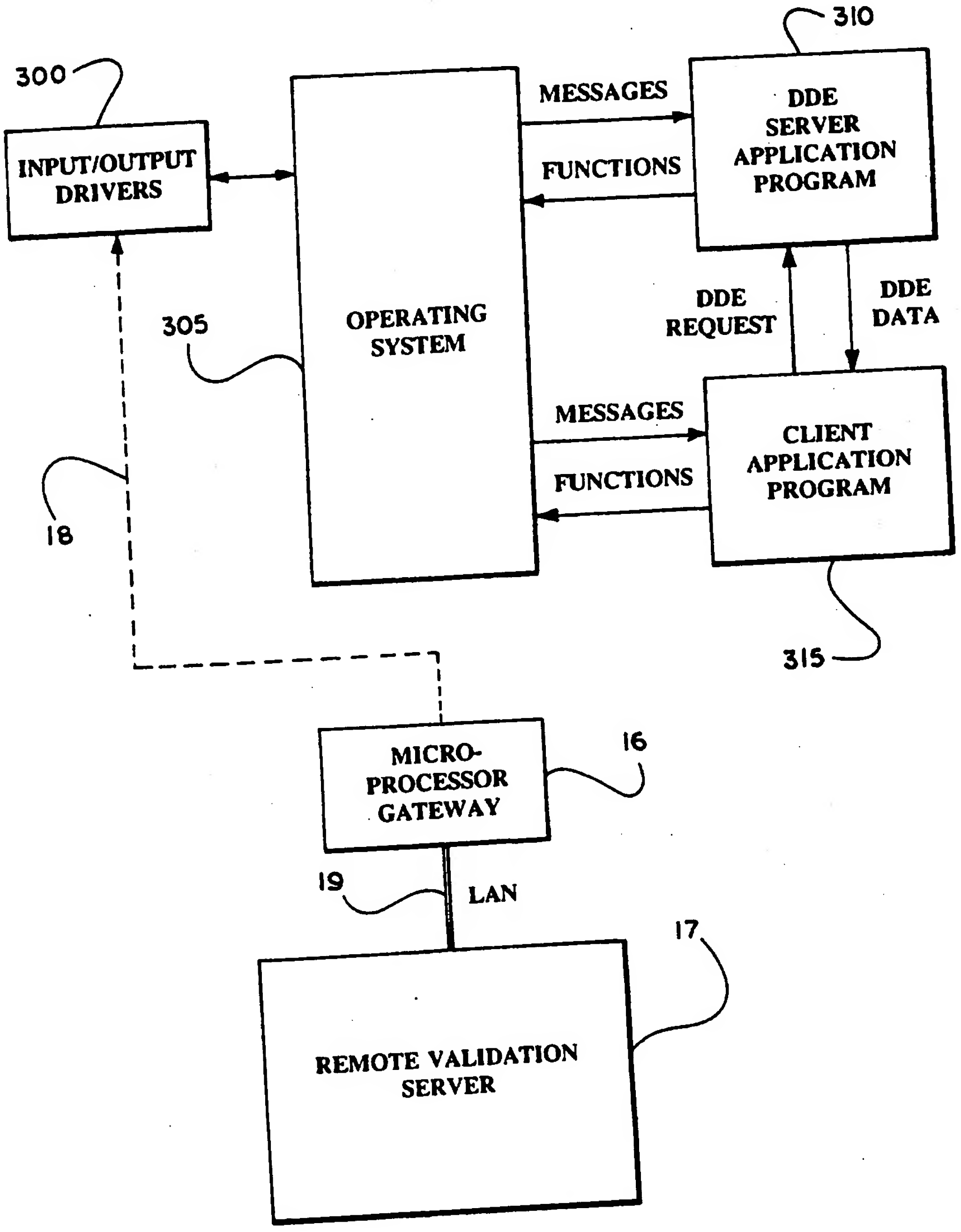

300
INPUT/OUTPUT DRIVERS
OPERATING SYSTEM
305
MESSAGES
FUNCTIONS
310
DDE SERVER APPLICATION PROGRAM
DDE REQUEST
DDE DATA
MESSAGES
FUNCTIONS
CLIENT APPLICATION PROGRAM
315
18
MICRO-PROCESSOR GATEWAY
16
19
LAN
17
REMOTE VALIDATION SERVER

Fig_3

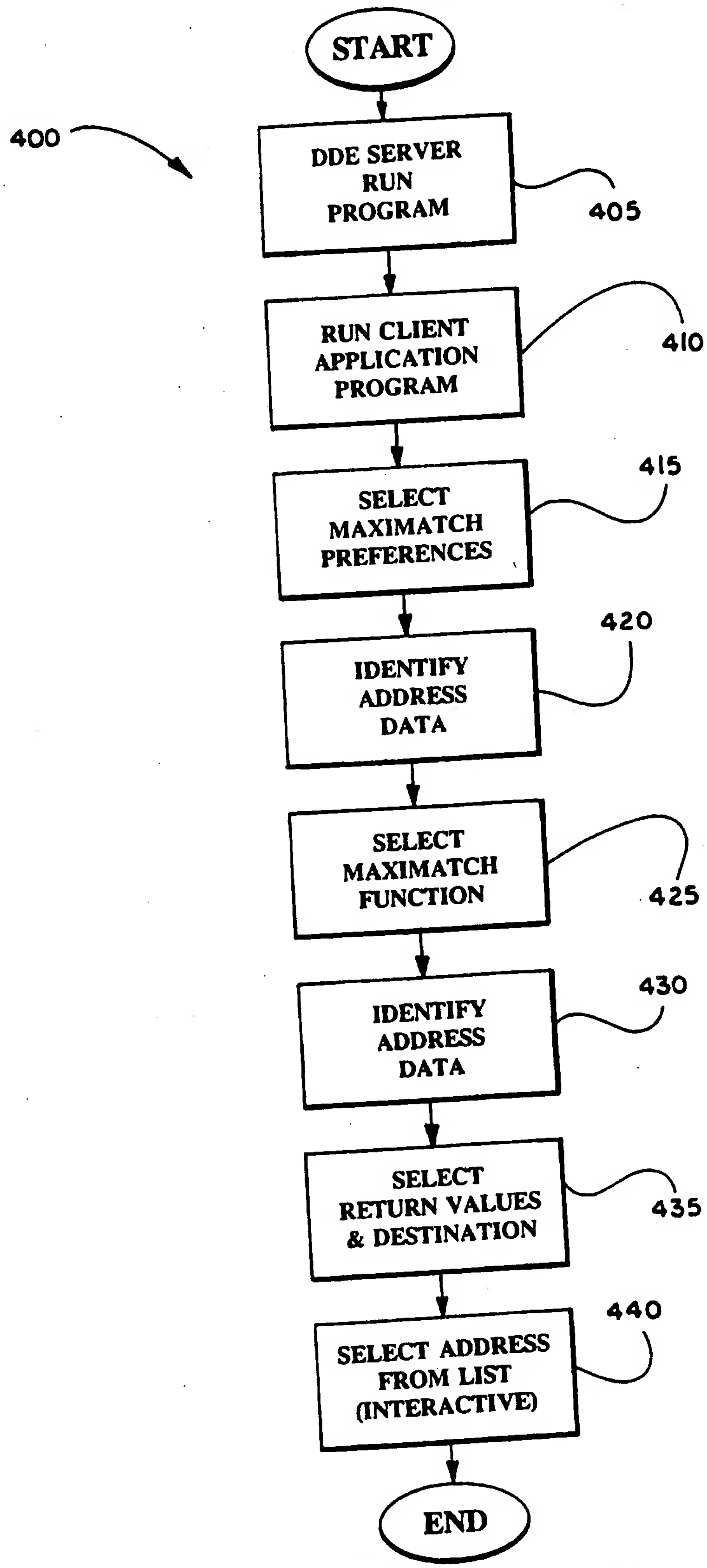

START
400
DDE SERVER RUN PROGRAM
405
RUN CLIENT APPLICATION PROGRAM
410
SELECT MAXIMATCH PREFERENCES
415
IDENTIFY ADDRESS DATA
420
SELECT MAXIMATCH FUNCTION
425
IDENTIFY ADDRESS DATA
430
SELECT RETURN VALUES & DESTINATION
435
SELECT ADDRESS FROM LIST (INTERACTIVE)
440
END

Fig_4

STATE DIAGRAM:
PROCESS PERFORMED BY CLIENT APPLICATION

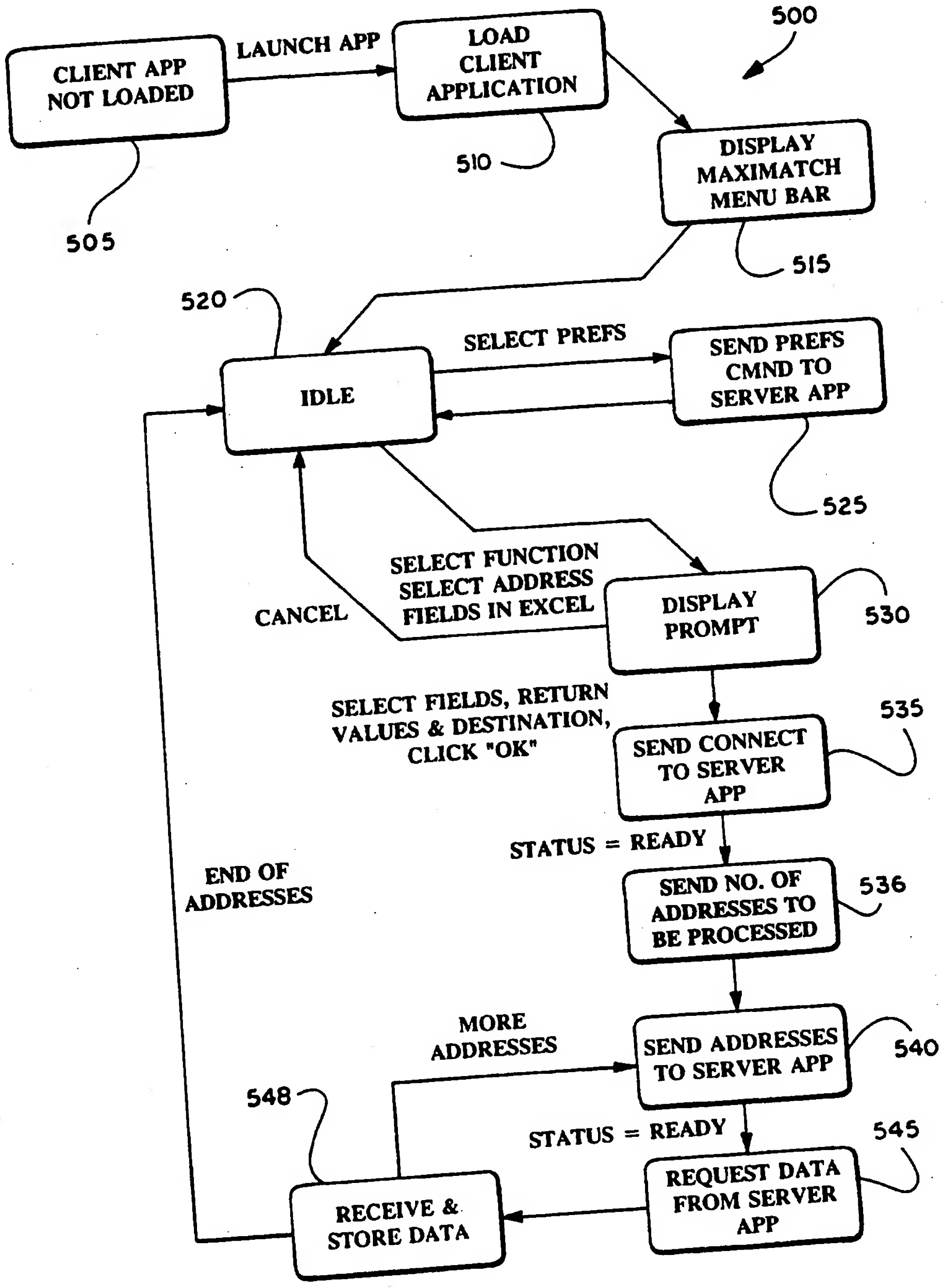

500
CLIENT APP NOT LOADED
505
LAUNCH APP
LOAD CLIENT APPLICATION
510
DISPLAY MAXIMATCH MENU BAR
515
520
IDLE
SELECT PREFS
SEND PREFS CMND TO SERVER APP
525
CANCEL
SELECT FUNCTION SELECT ADDRESS FIELDS IN EXCEL
DISPLAY PROMPT
530
SELECT FIELDS, RETURN VALUES & DESTINATION, CLICK "OK"
SEND CONNECT TO SERVER APP
535
STATUS = READY
END OF ADDRESSES
SEND NO. OF ADDRESSES TO BE PROCESSED
536
MORE ADDRESSES
SEND ADDRESSES TO SERVER APP
540
548
STATUS = READY
545
REQUEST DATA FROM SERVER APP
RECEIVE & STORE DATA

Fig_5

STATE DIAGRAM:
PROCESS PERFORMED BY DDE SERVER PROGRAM

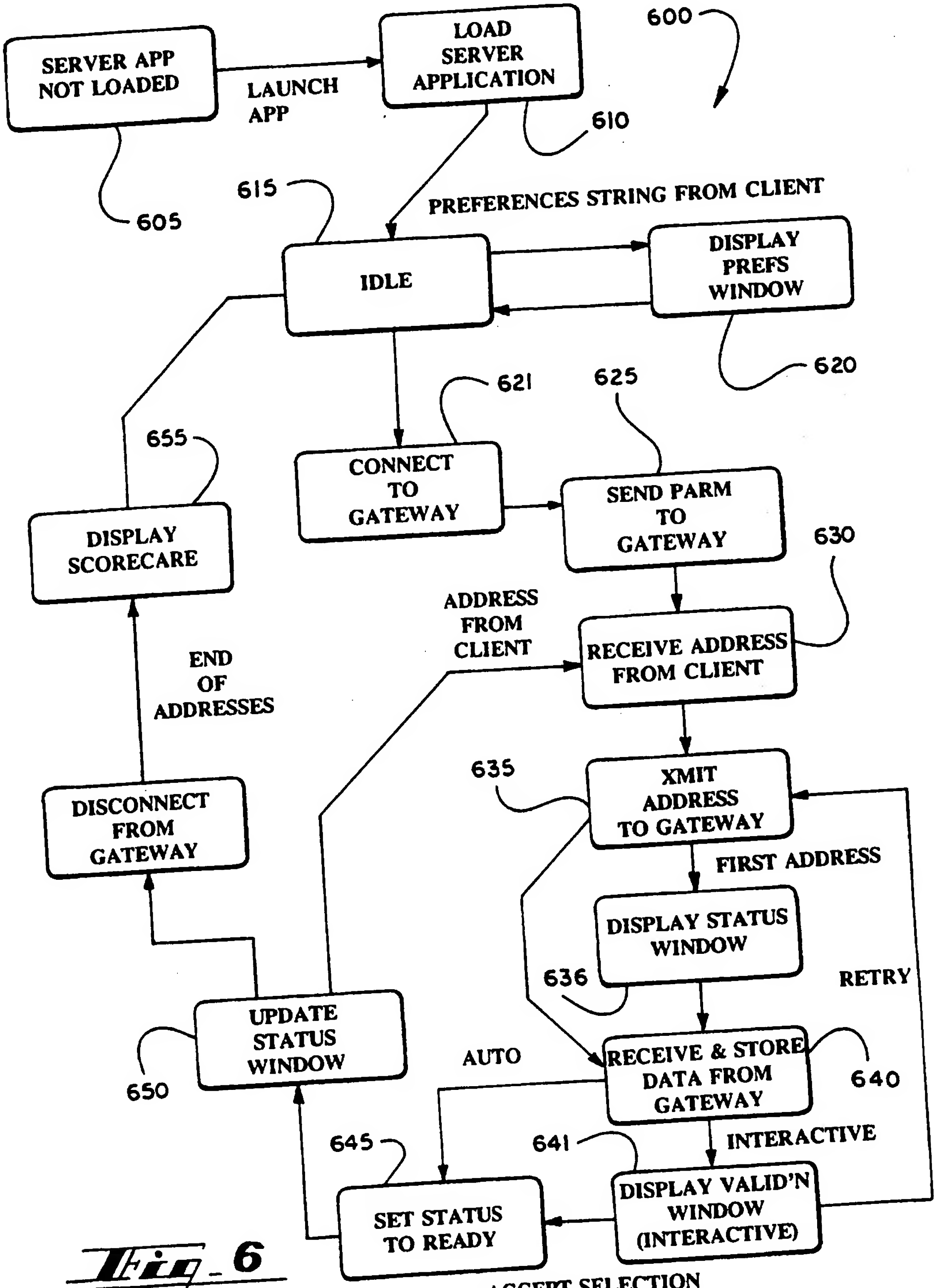


Fig_6

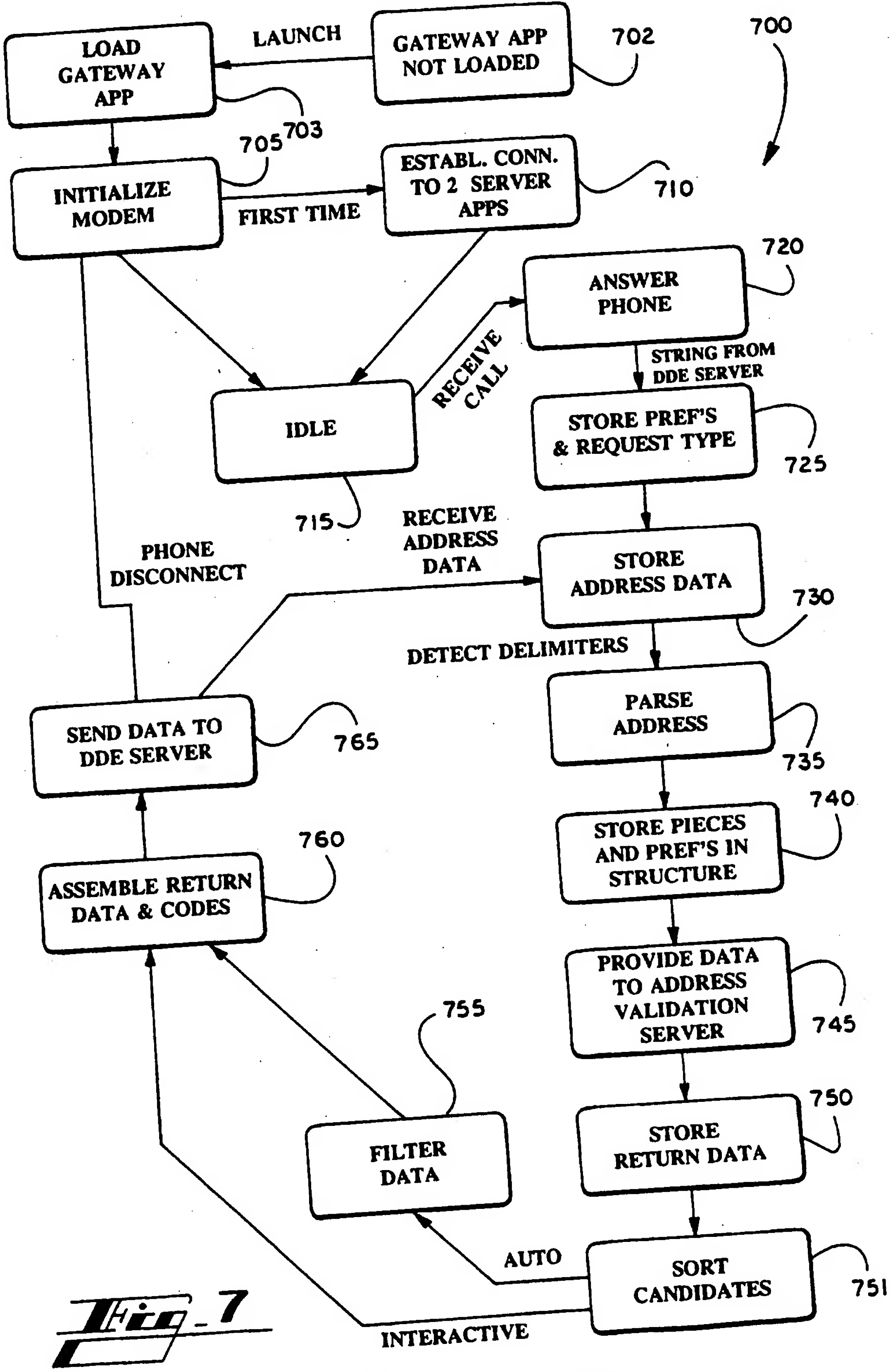

LOAD GATEWAY APP
LAUNCH
GATEWAY APP NOT LOADED
702
700
703
705
INITIALIZE MODEM
FIRST TIME
ESTABL. CONN. TO 2 SERVER APPS
710
720
ANSWER PHONE
RECEIVE CALL
IDLE
STRING FROM DDE SERVER
STORE PREF'S & REQUEST TYPE
725
715
RECEIVE ADDRESS DATA
PHONE DISCONNECT
STORE ADDRESS DATA
730
DETECT DELIMITERS
PARSE ADDRESS
735
SEND DATA TO DDE SERVER
765
740
STORE PIECES AND PREF'S IN STRUCTURE
760
ASSEMBLE RETURN DATA & CODES
PROVIDE DATA TO ADDRESS VALIDATION SERVER
745
755
750
STORE RETURN DATA
FILTER DATA
AUTO
SORT CANDIDATES
751
INTERACTIVE

Fig_7

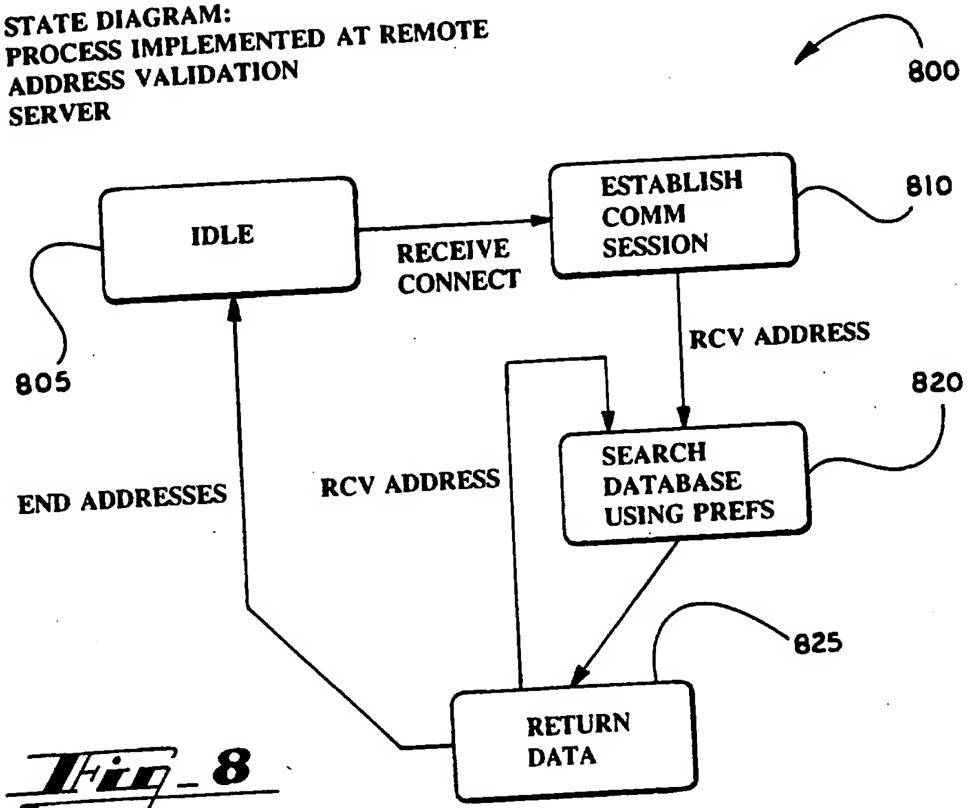

STATE DIAGRAM:
PROCESS IMPLEMENTED AT REMOTE
ADDRESS VALIDATION
SERVER
800
IDLE
805
RECEIVE
CONNECT
ESTABLISH
COMM
SESSION
810
RCV ADDRESS
820
SEARCH
DATABASE
USING PREFS
END ADDRESSES
RCV ADDRESS
825
RETURN
DATA

Fig_8

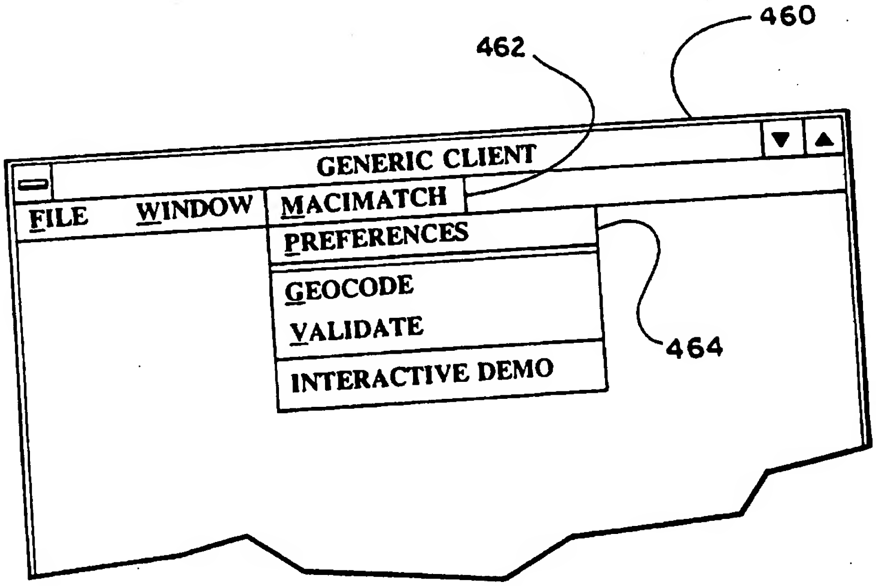
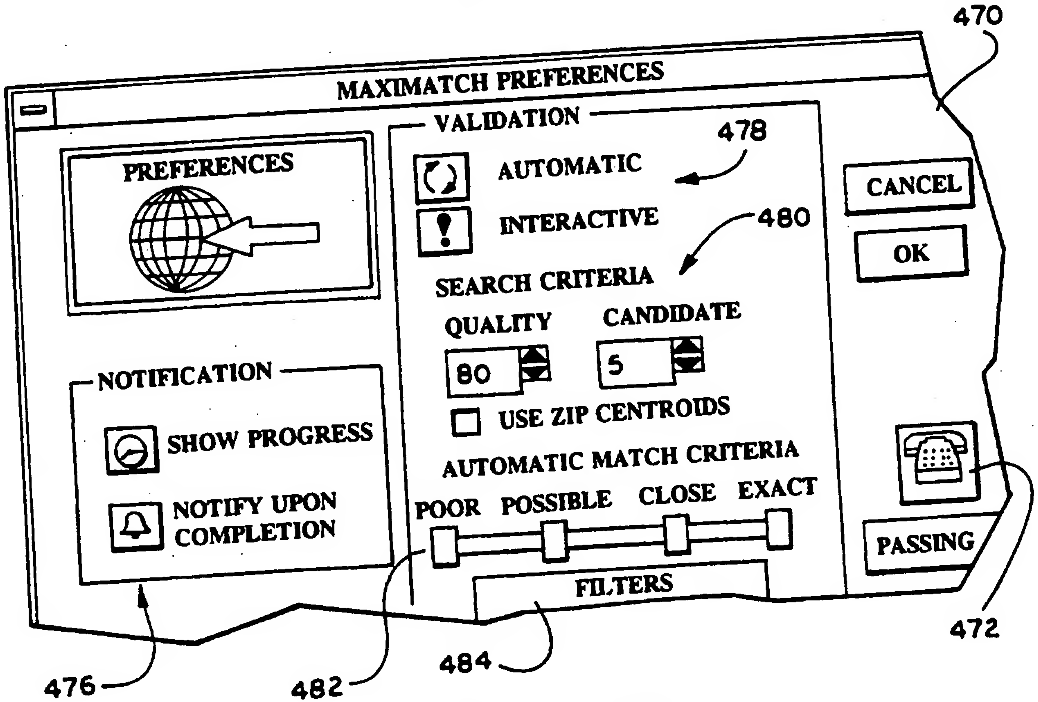

462
460
GENERIC CLIENT
FILE
WINDOW
MACIMATCH
PREFERENCES
GEOCODE
VALIDATE
INTERACTIVE DEMO
464

Fig_9



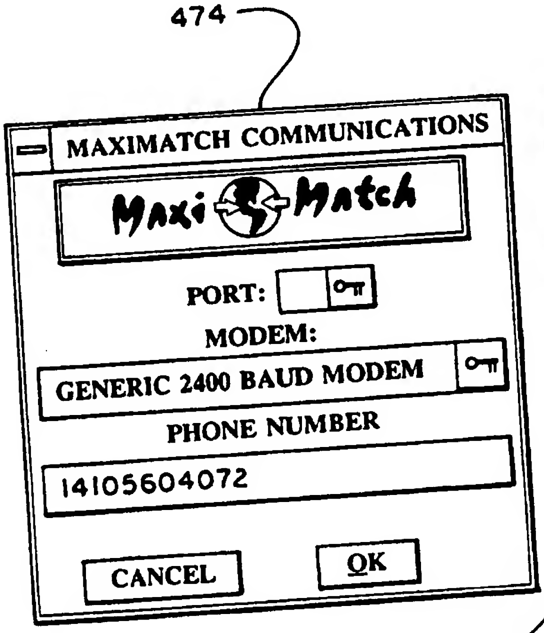
SEARCH CRITERIA

QUALITY 80

CANDIDATE 5

□ USE ZIP CENTROIDS

FIG_10

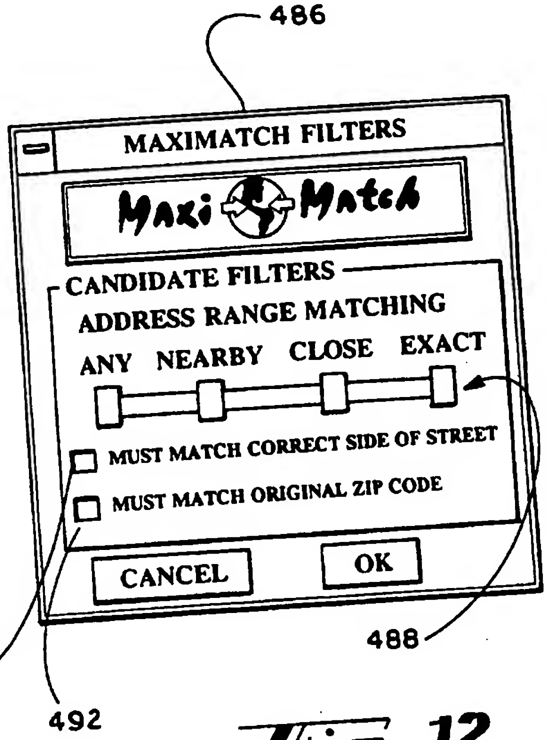


PORT:

MODEM: GENERIC 2400 BAUD MODEM

PHONE NUMBER 14105604072

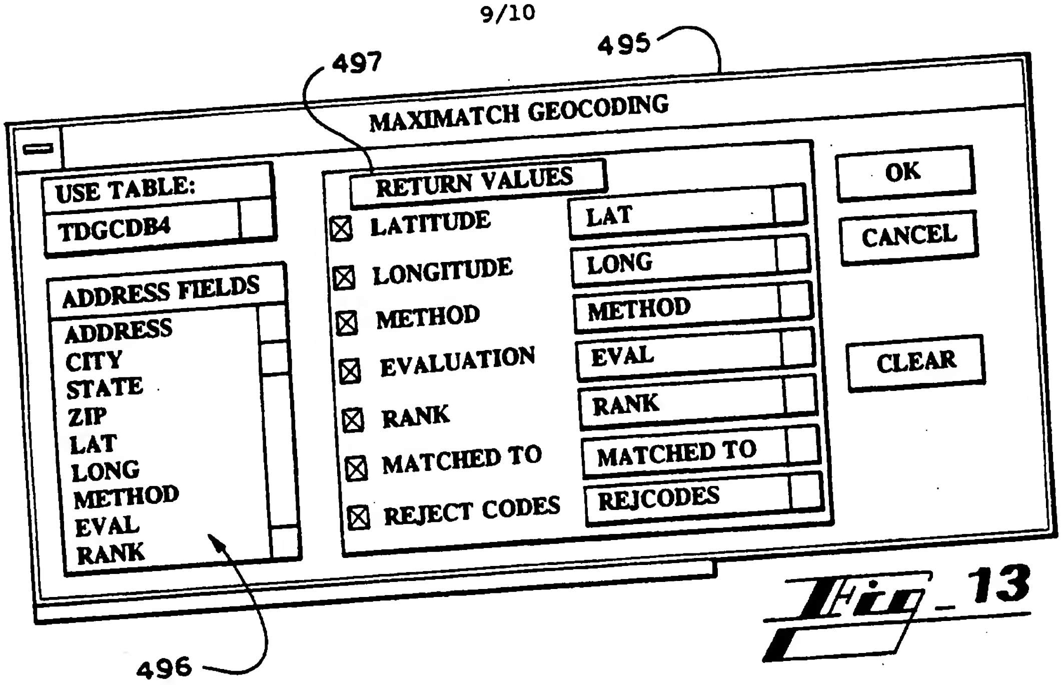
FIG_11



□ MUST MATCH CORRECT SIDE OF STREET

□ MUST MATCH ORIGINAL ZIP CODE

FIG_12

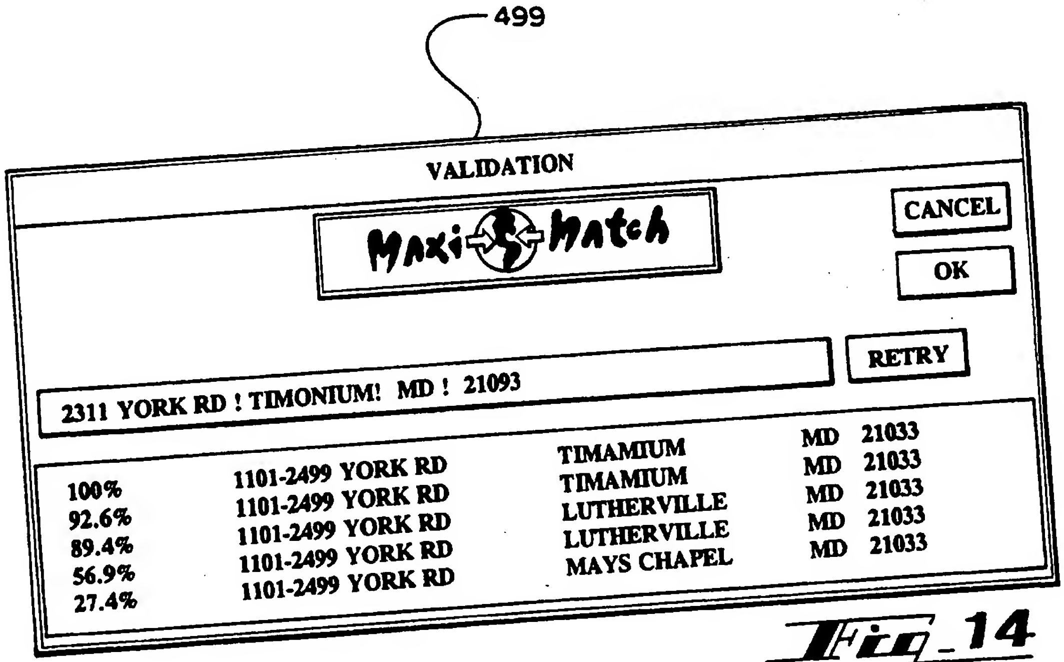


USE TABLE: TDGCDB4

RETURN VALUES

☒ LATITUDE	LAT
☒ LONGITUDE	LONG
☒ METHOD	METHOD
☒ EVALUATION	EVAL
☒ RANK	RANK
☒ MATCHED TO	MATCHED TO
☒ REJECT CODES	REJCODES

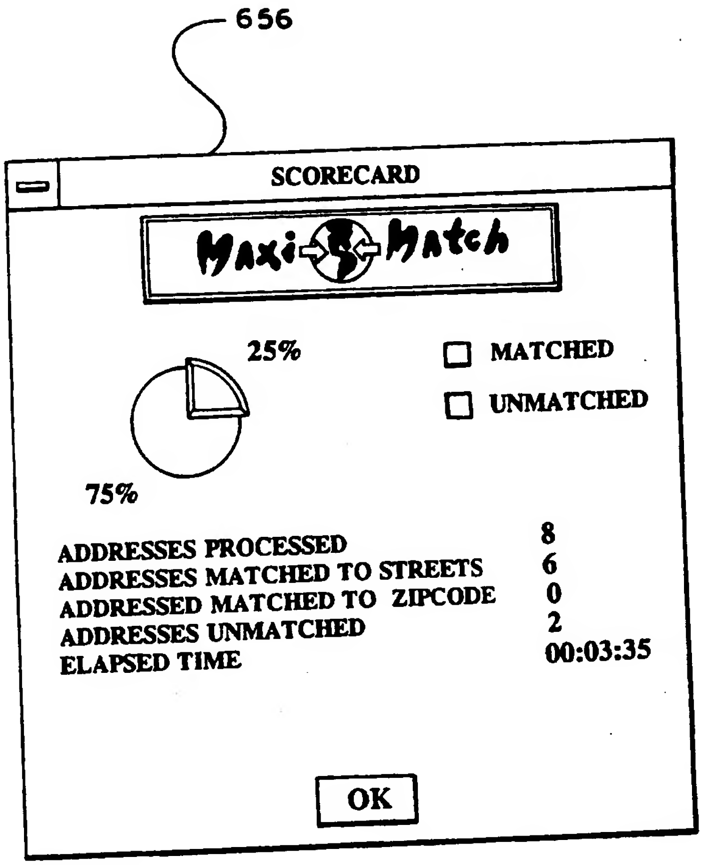
Fig_13



2311 YORK RD ! TIMONIUM! MD ! 21093

100%	1101-2499 YORK RD	TIMAMIUM	MD	21033
92.6%	1101-2499 YORK RD	TIMAMIUM	MD	21033
89.4%	1101-2499 YORK RD	LUTHERVILLE	MD	21033
56.9%	1101-2499 YORK RD	LUTHERVILLE	MD	21033
27.4%	1101-2499 YORK RD	MAYS CHAPEL	MD	21033

Fig_14



ADDRESSES PROCESSED	8
ADDRESSES MATCHED TO STREETS	6
ADDRESSED MATCHED TO ZIPCODE	0
ADDRESSES UNMATCHED	2
ELAPSED TIME	00:03:35

Fig_15

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 96/05734

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 G06F17/60 G06F17/30

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US,A,5 341 505 (WHITEHOUSE HARRY T) 23 August 1994 see column 6, line 25 - column 7, line 31 ---	1-7
A	WO,A,94 11810 (MICROSOFT CORP) 26 May 1994 see page 1, line 11 - line 27 ---	1,6,7
A	WHICH COMPUTER?, OCT. 1994, UK, ISSN 0140-3435, pages 46-47, 49 - 50, 53, 55, XP000578839 "Don't forget the postcode (addressing systems)" see the whole document -----	1-7

□ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search
13 August 1996

Date of mailing of the international search report
09.09.96

Name and mailing address of the ISA
European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer
Fournier, C

2

INTERNATIONAL SEARCH REPORT

Information on patent family members

Inte onal Application No
PCT/US 96/05734

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US-A-5341505	23-08-94	NONE	
WO-A-9411810	26-05-94	EP-A- 0669020	30-08-95
		US-A- 5511197	23-04-96